5-22-2006

Target Detection Using a Wavelet-Based Fractal Scheme

Gregory W. Stein
University of New Orleans

TARGET DETECTION USING A WAVELET-BASED FRACTAL SCHEME

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Engineering
Electrical

by

Gregory W. Stein

B.S. University of New Orleans, 2004

May, 2006

Dedication

This thesis is dedicated to my family and friends for their love and support.

Acknowledgments

I would like to express my gratitude towards my Major Professor, Dr. Dimitrios Charalampidis, for his guidance, technical knowledge, and encouragement throughout my graduate studies. I would like to acknowledge my committee members, Dr. Edit Bourgeois, Dr. Huimin Chen, and Dr. Vesselin Jilkov, whom I have the utmost respect. The Electrical Engineering department as The University of New Orleans has been a great place for my collegiate studies, and deserves acknowledgement. Lastly, I would like to acknowledge my family and friends. In particular, my parents, Calvin C. Stein, Jr., and Brenda S. Stein, for their unconditional love and support, and to my brother, Douglas C. Stein.

Contents

Abstract **vii**

List of Figures **viii**

1 Introduction **1**

1.1 SAR and ATR 1

1.2 Intensity-Based Features 2

1.3 Textural-Based Features 2

1.3.1 Previous Textural-Based Features 2

1.3.2 The Proposed Wavelet/Fractal Feature 3

1.4 Thesis Organization 3

2 Background **5**

2.1 Two Parameter CFAR 5

2.2 The Extended Fractal (EF) Feature 6

3 The Proposed Technique - Wavelet/Fractal Feature **8**

3.1 Wavelet Definition 8

3.2 Wavelet Transform, Roughness Feature,
and Structure Function 10

3.3 Average Weighted Roughness Feature 13

3.4 Filter Optimization 14

3.4.1 Method 1 14
3.4.2 Method 2 15
3.5 Advantages of AR Feature 15

4 Experimental Results **17**

5 Conclusions **23**

Bibliography **25**

Appendix **31**

Vita **31**

Abstract

In this thesis, a target detection technique using a rotational invariant wavelet-based scheme is presented. The technique is evaluated on Synthetic Aperture Rader (SAR) imaging and compared with a previously developed fractal-based technique, namely the extended fractal (EF) model. Both techniques attempt to exploit the textural characteristics of SAR imagery. Recently, a wavelet-based fractal feature set, similar to the proposed one, was compared with the EF feature for a general texture classification problem [1]. The wavelet-based technique yielded a lower classification error than EF, which motivated the comparison between the two techniques presented in this paper. Experimental results show that the proposed techniques feature map provides a lower false alarm rate than the previously developed method.

List of Figures

2.1 Annular window for calculation of CFAR feature 6

3.1 Wavelet illustration 9

3.2 Texture sample 11

3.3 Directional roughness feature illustration 12

3.4 (a) Feature values around a target point, and (b) impulse response of matched filter 13

4.1 Experimental Result 1 20

4.2 Experimental Result 2 21

4.3 Experimental Result 3 22

Chapter 1

Introduction

1.1 SAR and ATR

Synthetic Aperture Radar (SAR) is commonly used to obtain ground images for several remote sensing and military related applications. In particular, one of the most important military applications is Automatic Target Recognition (ATR). A successful ATR system should be able to identify targets and distinguish them from clutter objects, which may include grass, trees, buildings, or other structures. This task is usually referred to as target detection. In addition, an ATR system should be able to recognize, or identify, the detected targets. Target recognition is, in general, computationally expensive. Consequently, it may be impractical to attempt identifying and recognizing objects using all the image information provided by the SAR system. It is therefore important to first apply a target detection technique in order to identify image areas that include candidate targets with high probability. Once the image search space is significantly reduced, more sophisticated algorithms can be used for target recognition.

1.2 Intensity-Based Features

Constant False Alarm Rate (CFAR) target detectors [2–8] have been developed in order to take advantage of the fact that the radar cross section of a target is higher than the surrounding clutter. The two-parameter CFAR feature [2] is a computationally inexpensive clutter adaptive statistic, and for this reason is considered to be one of the benchmark algorithms for target detection. It assumes that the background clutter follows a Gaussian distribution. Other CFAR detectors have been developed [3–7] that consider more realistic clutter model distributions, such as Weibull. Rifkin [5] focuses on order statistic-based algorithms which calculate radar detection thresholds. These determine closed-form approximations for the signal-to-clutter ratio required to achieve a particular probability of detection in clutter environments dominated by the Weibull distribution. Morgan et al [7] presents a technique for determining the ideal detection threshold when Gaussian noise and Weibull distributed clutter returns are present on a radar receiver and neither is dominant. Kaplan [9] mentions several techniques that search for objects only on the basis of contrast or intensity [10] may be problematic. In particular, several clutter objects included in SAR imagery (such as trees and buildings) may have intensities similar to that of the true targets.

1.3 Textural-Based Features

1.3.1 Previous Textural-Based Features

As an alternative to CFAR-based features, texture-based features for SAR imagery have been investigated. Essentially, these techniques view the clutter portrayed in SAR images as a textural surface. Ideally, a successful feature set identifies local textural differences between target and clutter regions. Subotic [11] uses differences in the phase information from targets and clutter. Pham [12] combines a local variance feature with more traditional contrast and brightness features. Steward [13] uses fractal random process models for analysis and

segmentation of clutter in high-resolution SAR imagery. The fractal dimension of natural clutter sources was used as an input to a Bayesian classifier. Scenes were segmented into three classes, namely shadows, trees, and grass. The relevance of the resulting segmentation maps to CFAR target detection techniques was also discussed. Kaplan [9] introduces a new feature, namely the Extended Fractal (EF) feature, to improve target detection performance. The EF feature is similar to the multiscale Hurst parameter used to measure multiscale texture roughness [14]. Comparisons illustrated that EF can provide a smaller number of false alarms than the contrast-only features, such as the two-parameter CFAR, for a specific detection rate.

1.3.2 The Proposed Wavelet/Fractal Feature

In this thesis, a wavelet-based fractal feature set for target detection is proposed. Both EF and the proposed feature sets attempt to exploit the textural characteristics of SAR imagery. However, Charalampidis has recently shown that a Wavelet/Fractal (WF) feature set [1], similar to the proposed one, provided lower classification error rates than a feature set similar to EF for a general texture classification problem. The promising performance of the WF features motivated the comparison presented in this thesis between the two techniques. It should be stressed that this thesis is not a direct implementation of Charalampidis' WF feature [1]; furthermore, a heavily modified partition function is used for this SAR application, discussed in Chapter 3. Thus, this thesis concentrates on the development of a purely textural feature used to detect targets in SAR images.

1.4 Thesis Organization

This thesis is organized as follows. In Chapter 2, background information regarding the CFAR and EF features is presented. Chapter 3 presents the author's feature extraction

scheme. Chapter 4 provides comparisons of the EF and WF features. Finally, conclusions and discussions about future work are presented.

Chapter 2

Background

In this chapter, the two-parameter CFAR and the EF feature are described. The first, CFAR, is presented since it is a commonly used feature for target detection, while the second, EF, can be considered a predecessor to the author's feature set.

2.1 Two Parameter CFAR

The two-parameter CFAR feature used in SAR ATR systems [2] attempts to distinguish targets from clutter based on the idea that the image intensity corresponding to targets should be larger than that of the surrounding clutter. The feature is computed for each image pixel. In addition, CFAR assumes the background clutter follows a Gaussian distribution. Considering a pixel located at (m, n), CFAR is defined as the difference between the pixel value $I(m, n)$ and the surrounding clutter normalized by the surrounding clutter standard deviation:

$$C(m, n) = \frac{I(m, n) - \hat{\mu}(m, n)}{\hat{\sigma}(m, n)} \tag{2.1}$$

The values $\hat{\mu}(m, n)$, the clutter mean, and $\hat{\sigma}(m, n)$, the clutter standard deviation, are estimated over a one pixel-wide square annular window containing N_c elements around location (m, n) as shown in Figure 2.1. The radius of the annular window should not be smaller than

the length of the target to ensure that the window only includes pixels corresponding to clutter. The previous statement implies that some prior knowledge about the target's size (in pixels) is required for this feature.

To illustrate the two-parameter CFAR feature, assume a target is within the annular win-

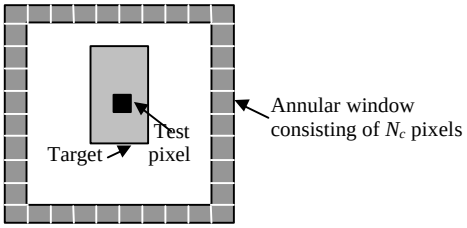


Figure 2.1: Annular window for calculation of CFAR feature

dow. Based on the assumption that targets have higher intensity than surrounding clutter, $I(m,n) - \hat{\mu}(m,n)$, normalized by $\hat{\sigma}$, outputs a pixel with relatively high intensity. Thus, in the resultant image, the target locations have high intensity, and all surrounding pixels have much lower intensity. The CFAR algorithm presented in this section does not always perform as expected; in fact, it is accepted that CFAR produces a high false alarm rate, but it still considered a benchmark idea. However, the processing time for the CFAR algorithm is minimal.

2.2 The Extended Fractal (EF) Feature

The EF feature, unlike the CFAR feature, attempts to exploit the textural characteristics of the SAR images. Given a discrete image $I(m,n)$, the x- and y-directed EF features are defined as the log ratio of the local average power considering lags 2Δ and 4Δ in the x- and y-directions, respectively:

$$F^x(m,n) = \frac{1}{2}\log_2 \frac{f^x_\Delta(m,n)}{f^x_{2\Delta}(m,n)}$$
$$F^y(m,n) = \frac{1}{2}\log_2 \frac{f^y_\Delta(m,n)}{f^y_{2\Delta}(m,n)} \tag{2.2}$$

where f^x and f^y are called structure functions defined as:

$$f^x_\Delta(m,n) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} |I(m+\Delta+i, n+j) - I(m-\Delta+i, n+j)|^2$$

$$f^y_\Delta(m,n) = \sum_{i=-w}^{w} \sum_{j=-w}^{w} |I(m+i, n+\Delta+j) - I(m+i, n-\Delta+j)|^2 \tag{2.3}$$

As Equation 2.3 describes, the structure functions are computed considering a sliding window of size $W \times W$, where $W = 2w + 1$ The window size W depends on the smallest lag so that

$$\Delta = \frac{W-1}{4}. \tag{2.4}$$

Note that Δ must be an integer so that the window size must equal one plus a multiple of four. To achieve directional invariance, the EF feature is defined as the average of the two directed features.

$$F(m,n) = \frac{F_x(m,n) + F_y(m,n)}{2} \tag{2.5}$$

Chapter 3

The Proposed Technique - Wavelet/Fractal Feature

In this chapter, a new feature extraction technique used for target detection is presented. In this feature, fractal measures are computed using directional wavelets defined in the following sections [1].

3.1 Wavelet Definition

A wavelet, which is part of a larger fractal family, is simply a mathematical function that segments data into different components at a given scale s. For the purposes of this thesis, the wavelet needs to be viewed as a transition detector in the SAR images.
A two dimensional Gaussian smoothing function at scale s is defined:

$$\Phi(x, y, s) = e^{-\frac{x^2+y^2}{2s^2}} \tag{3.1}$$

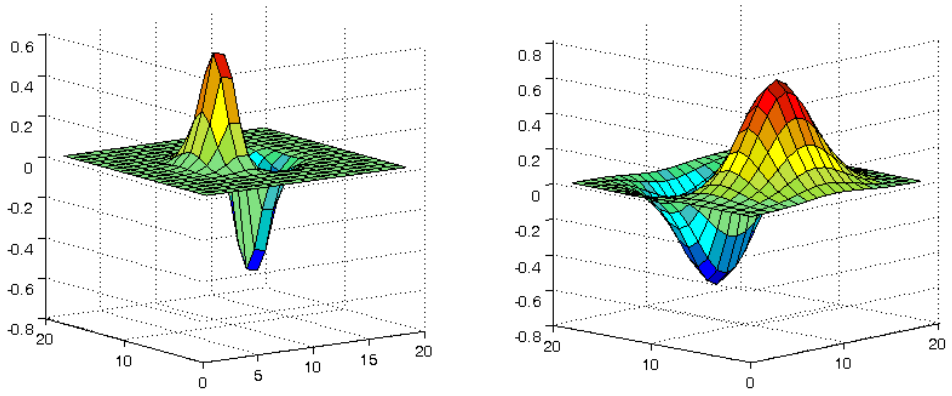
Two directional wavelets, at 0° and 90°, are defined in Equation 3.2 as the partial derivatives of the smoothing function $\Phi(x, y, s)$ along x- and y-directions, respectively. Note that the

subscript indicates the wavelet direction angle.

$$W_0(x,y,s) = \frac{\partial \Phi(x,y,s)}{\partial x} = -\frac{x}{s} e^{-\frac{x^2+y^2}{2s^2}}$$

$$W_{90}(x,y,s) = \frac{\partial \Phi(x,y,s)}{\partial y} = -\frac{y}{s} e^{-\frac{x^2+y^2}{2s^2}} \tag{3.2}$$

A simple illustration for the 0°-direction at scale 1.5 and 90°-direction at scale 3 is shown in Figure 3.1:



(a) 0 ° Wavelet – Scale 1.5 (b) 90 ° Wavelet – Scale 3

Figure 3.1: Wavelet illustration

One may notice that the wavelets in Figure 3.1 are not normalized, indicating that the wavelet at scale 3 has much higher energy than the wavelet at scale 1.5. However, this phenomena does not need to be accounted for since computations only occur using one scale at a time.

Given an image $I(x,y)$, and because differentiation and convolution are linear operators:

$$W_0(x,y,s) * I(x,y) = \frac{\partial \Phi(x,y,s)}{\partial x} * I(x,y) = \frac{\partial}{\partial x}[\Phi(x,y,s) * I(x,y)]$$

$$W_{90}(x,y,s) * I(x,y) = \frac{\partial \Phi(x,y,s)}{\partial y} * I(x,y) = \frac{\partial}{\partial y}[\Phi(x,y,s) * I(x,y)] \tag{3.3}$$

Therefore, $W_0(x,y,s) * I(x,y)$ and $W_{90}(x,y,s) * I(x,y)$ are the gradient components of $\Phi(x,y,s) * I(x,y)$ in directions 0° and 90°, or the filtered versions of $I(x,y)$ using filters $W_0(x,y,s)$ and $W_{90}(x,y,s)$, respectively. Computation of the filtered image in an arbitrary direction θ requires filtering the image $I(x,y)$ by the wavelet $W_\theta(x,y,s) = W_0(xcos\theta + y\sin\theta, -\sin\theta + y\cos\theta, s)$. The convolution $W_\theta(x,y,s) * I(x,y)$ is the θ-directed gradient component of $\Phi(x,y,s) * I(x,y)$ along $x^{'} = x\cos\theta + y\sin\theta$ and can be computed as a linear combination of the 0° and 90° components:

$$W_\theta * I(x,y) = [W_0(x,y,s) * I(x,y)]\cos\theta + [W_{90}(x,y,s) * I(x,y)]\sin\theta \tag{3.4}$$

Therefore, $W_\theta(x,y,s)$ is said to be steerable, and $W_\theta(x,y,s) * I(x,y)$ is computed for different angles of θ from the 0° and 90° components using Equation 3.4. The steerability of the wavelet saves significantly in computation time because only two convolutions need to be computed for any filtering direction. As shown by Charalampidis [1], the second partial derivatives as well as higher order derivatives of the smoothing function can be considered; however, for the purposes here, only first order derivatives are considered.

3.2 Wavelet Transform, Roughness Feature, and Structure Function

Consider the following wavelet transform of image $I(x,y)$ at scale s and direction θ:

$$WT_I^\theta(x,y,s) = W_\theta(x,y,s) * I(x,y) \tag{3.5}$$

Next the directional roughness features $R_s^\theta(x,y)$ are determined for every pixel by the following power-law relation of the partition function $\mu_{s,\theta,\epsilon,N}(x,y)$:

$$\mu_{s,\theta,\epsilon,N}(x,y) = \left\langle max_{(u,v)\in A_\epsilon(x,y)} \left|WT_I^\theta(u,v,s)\right| - min_{(u,v)\in A_\epsilon(x,y)} \left|WT_I^\theta(u,v,s)\right| \right\rangle_{N\times N} \tag{3.6}$$

where $<>_{N\times N}$ indicates the spatial arithmetic average in a $N \times N$ window and A_ϵ is defined as an annular window of radius ϵ, similar to the one shown in Figure 2.1. In essence, the partition function is finding the difference between the maximum and minimum value in an annular window for each pixel in $WT_I^\theta(x, y, s)$. The directional roughness feature for each image pixel is computed as the slope of the line that best fits $(\log \epsilon, \log \mu_{s,\theta,\epsilon,N}(x, y))$. This implies that ϵ is a vector considered for every θ. The partition function $\mu_{s,\theta,\epsilon,N}(x, y)$ of Equation 3.6 is a modified version of the one used by Charalampidis in [1]. Figure 3.3 illustrates examples of such lines for the structure function originally proposed by Charalampidis [1] for two different textures, similar to those displayed in Figure 3.2, at different texture locations, for $s = 2, N = 9, \theta = 0°$, and $\epsilon = [1\ 2\ 3]$. Notice that lines extracted from different locations

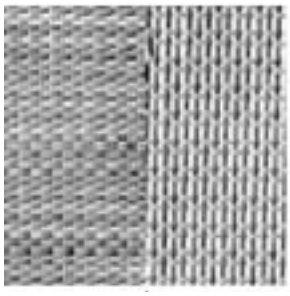


Figure 3.2: Texture sample

of the same texture are almost parallel to each other, which verifies the robustness of the directional roughness features, $R_s^\theta(x, y)$. The distance between lines corresponding to the same texture can be accounted for due to contrast differences at various texture locations. Furthermore, the directional roughness feature is less sensitive to intensity and contrast differences within the same texture, which is advantageous when attempting to distinguish targets from clutter in SAR images.

Next, some intuition is provided to help understand the reasoning behind this particular partition function. First, it should be emphasized that the transform of Equation 3.5 is the θ-direction gradient component of the image $I(x, y)$. This implies that high intensities in $WT_I^\theta(x, y, s)$ correspond to sharp transitions in $I(x, y)$. Thus, in general, transform

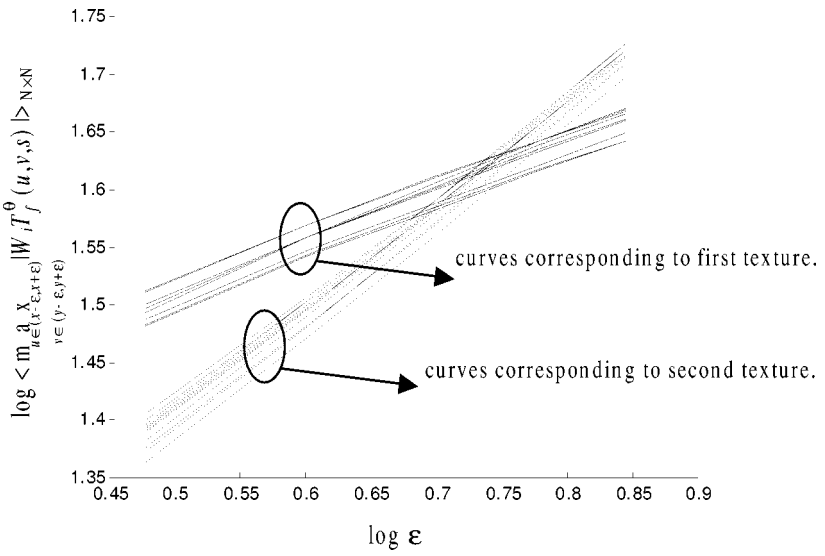


Figure 3.3: Directional roughness feature illustration

$WT_I^{\theta}(x, y, s)$ tends to highlight the target outlines, since moving from a target pixel to a background/clutter pixel presents a sharp transition. At the same time, transform pixels inside and outside the target possess a low intensity. The partition function is essentially the average difference, in a $N \times N$ window, between the maximum and minimum intensities of the transform in an annular window A_ϵ of radius ϵ.

Let us consider the case where a pixel is located inside the target. If the value of ϵ is similar to the target's radius, then A_ϵ includes pixels from the outline, as well as from inside the target, resulting in a large max-min difference. As ϵ increases A_ϵ includes pixels only from outside the target, resulting in a small max-min difference. Then, it is expected that, for pixels located inside the target, the max-min difference is decreasing as ϵ increases. This results in a negative exponent based on the definition of Equation 3.6. On the other hand, for pixels outside but relatively near the target's outline, the difference is small for small ϵ, since target outline pixels are not included in A_ϵ, and increases as ϵ increases, since A_ϵ includes more of the high valued outline pixels. In this case, the exponent of Equation 3.6 is positive. As a result, the feature inside and around the target area has a form as the one shown in Figure 3.4(a). In Figure 3.4(a), the negative of the feature value has been

considered for illustration purposes. Therefore, Figure 3.4(a) depicts a high negative value at the target surrounded by moderate positive values around the target.

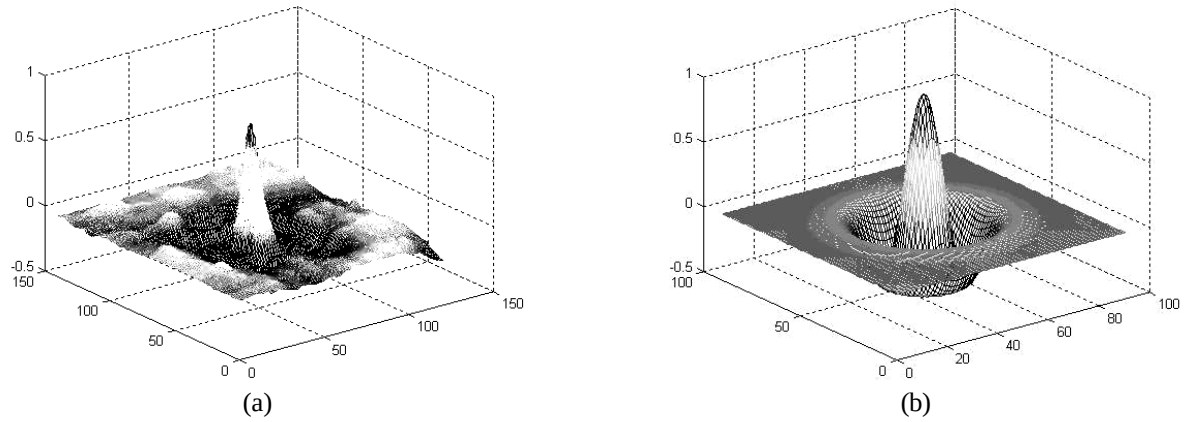


Figure 3.4: (a) Feature values around a target point, and (b) impulse response of matched filter

3.3 Average Weighted Roughness Feature

Due to the visual effect of roughness being highly dependent on the relative textural energy between different directions, the roughness features, $R_s^\theta(x, y)$, described in the previous section, must be weighted. For example, if a texture is rougher in direction θ_1 than direction θ_2, but has less energy, direction θ_1 may appear to be less rough than direction θ_2. As a result, Equation 3.7 describes a *percentage of energy* [1] feature computed in direction θ and scale s. The percentage of energy, Per_s^θ, is insensitive to both absolute image illumination since the DC component has been removed by the exponential wavelet and contrast changes since a change in contrast will multiply E_s^θ and E_s^{Total} by a common term.

$$Per_s^\theta = \frac{E_s^\theta}{E_s^{Total}} \tag{3.7}$$

where

$$E_s^\theta = \left\langle \left| WT_I^\theta(x, y, s) \right| \right\rangle_{NxN} \tag{3.8}$$

$$E_s^{Total} = \sum_{\theta} \left\langle \left| WT_I^{\theta}(x, y, s) \right| \right\rangle_{NxN} \tag{3.9}$$

Thus, a *weighted roughness* feature can be defined as the roughness feature $R_s^{\theta}(x, y)$ weighted with the percentage of energy with the same scale and direction:

$$WR_s^{\theta} = R_s^{\theta} Per_s^{\theta} \tag{3.10}$$

Finally, to obtain a rotationally invariant feature, an *average weighted roughness* feature is introduced:

$$AR_s = \frac{1}{Q} \sum_{\theta} WR_s^{\theta} \tag{3.11}$$

where Q is the total number of directions considered.

3.4 Filter Optimization

3.4.1 Method 1

As stated in Section 3.2, the general shape of a target's roughness feature is known to roughly follow the Gabor function; thus, a filter of the form,

$$h(x, y) = \cos(\omega_0 \sqrt{x^2 + y^2}) e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.12}$$

illustrated previously in Figure 3.4 can be considered a matched filter to the target. Figure 3.4(a) is the resultant image when the author's feature is applied to a T72 battle tank from the MSTAR database. Notice that this roughly follows the shape of Figure 3.4(b), which was created using Equation 3.12 with $\omega_0 = 0.06\pi$ and σ at the desired scale. These images can be used to enhance the feature space by emphasizing target parts by filtering AR_s with $h(x, y)$.

3.4.2 Method 2

Assuming that training data is available and an array of possible targets is known, an improved optimization technique can be utilized. Given target images, $T(x, y)$, an average can computed at scale s for all possible targets in a particular clutter image described by:

$$h(x, y) = \frac{1}{P} \sum_{T} AR_s^T(x, y) \tag{3.13}$$

where P is the total number of targets considered. It should be noted that for the experimental results presented in this thesis, Equation 3.13 is used since the targets in a particular clutter image are known. However, Equation 3.12 is also applied and performs quite well as a matched filter.

3.5 Advantages of AR Feature

The advantages of the author's AR feature over the EF feature are discussed herein. First, AR uses a partition function $\mu_{s,\theta,\epsilon,N}(x, y)$ that employs the smoothing function of Equation 3.1. Therefore, $\mu_{s,\theta,\epsilon,N}(x, y)$ is less sensitive to noise than other techniques, such as CFAR, that do not use any type of smoothing. Second, $\mu_{s,\theta,\epsilon,N}(x, y)$ is computed using steerable filters, and thus, roughness features can be computed in several directions with a relatively small computational overhead. As a result, it is expected that the average roughness features are less sensitive to target rotations compared to the EF features, which are computed as the average of only two directed features. Third, the proposed features are designed to detect target-like objects that are characterized by sharp transitions. This is a result of using the gradient-based filter. Finally, it appears that the proposed features provide better spatial resolution capabilities. More specifically, the feature map $AR_s(x, y)$ is an image of the same size as the original image $I(x, y)$; the EF features appear as a blob around the target's location, while the proposed features appear as a smaller spot inside the target's outline.

Furthermore, the AF feature outperforms the EF feature in the detection of closely spaced targets. For the reasons listed above, the AF feature is more robust to contrast changes and noise while maintaining the needed level of resolution to successfully detect target locations.

Chapter 4

Experimental Results

In this chapter, comparisons between the proposed AR feature and the EF feature are presented. The comparisons are performed primarily in terms of the ability to visualize the difference between the feature values corresponding to targets and clutter in the feature-map images. Furthermore, a quantitative, statistical approach has not been used to determine the percentage of targets detected.

A number of example SAR images have been tested. These images consist of a mixture of clutter and targets, obtained from the MSTAR database. The targets, including their shadows, are artificially inserted into the clutter images using Adobe Photoshop. Thus, the target locations are known. It should be noted that Kaplan has used a similar technique in [9]. Three of the examples for which the two features were compared are shown in Figures 4.1, 4.2, and 4.3. From these figures, it can be observed that the images appear to be realistic. For Figures 4.1, 4.2, and 4.3, (a) is the original image, and (b) is the original image with target markers. Figures (c) and (d) portray the feature maps for the AR and EF features respectively.

At this point, it should be emphasized that the goal of the detection technique is to re-

duce the false alarm rate. However, it is not expected that the percentage of false alarms can be reduced down to zero. In order to achieve a near-zero false alarm rate, a subsequent target/clutter recognition stage is necessary. If performed, the recognition step would attempt to classify targets and clutter into different categories and at the same time to eliminate any remaining false alarms. Nevertheless, since the recognition stage is time consuming, it is important to be able to reduce the target search space. This can be achieved by a target/clutter detection process such as the one proposed in this work.

For the three subsequent comparisons, the following AR feature parameters were applied on an image downsampled at a 2:1 ratio:

- $N = 19$
- $\epsilon = [7\ 11\ 15]$
- $\theta = [0^\circ\ 45^\circ\ 90^\circ\ 135^\circ]$
- $s = 1.5$

Similarly, the following EF feature parameters were applied on the same downsampled images:

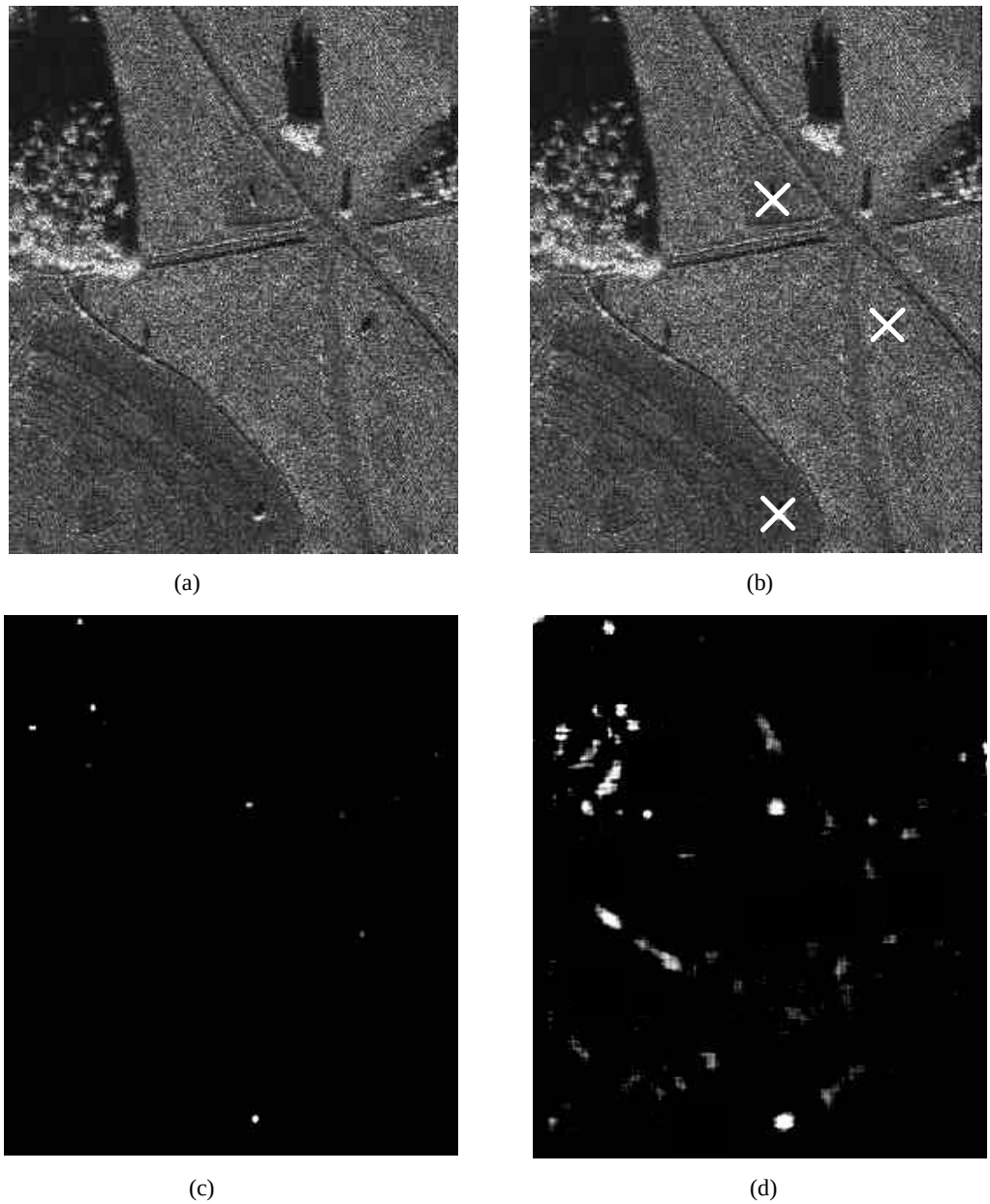
- $\Delta = 7$
- $W = 4\Delta + 1 = 29$

From all three examples presented, it can be observed that the AR feature can be used to relatively easily identify the target locations. More specifically, as it is illustrated in Figure 4.1(c), the AR feature possesses high values for locations in the feature maps associated to the targets. There are some other points in the AR feature map that also possess relatively high AR feature values, which could be mistaken to represent targets in the original image. However, it can be observed from Figure 4.1(a) that the objects that exist at the corresponding points in the original image do have a target-like appearance. Therefore, having relatively

high AR feature values at the corresponding locations in the feature map should be expected.

Figure 4.1(d) depicts the EF feature map. It is apparent that the EF feature may result in more potential false alarms compared to the AR feature. Although the target locations correspond to relatively large EF feature values in the EF feature map, there are many clutter-associated locations with significantly high EF feature values. In particular, it can be seen in Figure 4.1(d) that the rightmost target may not be easily detected using a thresholding technique, since there are many other locations in the feature map with significantly higher EF values that do not actually correspond to a target. It should be mentioned that the EF features maps were contrast-enhanced in order to emphasize large feature values for illustration purposes. The example presented in Figure 4.2 may result in conclusions similar to Figure 4.1. In this case, AR clearly shows the 4 target locations, while EF shows a significant number of potential false alarms.

The comparison for Figure 4.3 displays the improved spatial resolution capability of the AR feature compared with the EF feature. In the upper right of the original image, three closely spaced targets are marked. In the resultant feature maps (which has the area in question boxed in grey), it is clear that the EF feature results in one large blob encompassing all three targets. On the other hand, the AR feature distinguishes all three targets. Figure 4.3 also illustrates the robustness of the AR feature in terms of its insensitivity to contrast in clutter. The large mass of trees in the EF feature map displays far more false alarms than the AR feature map.

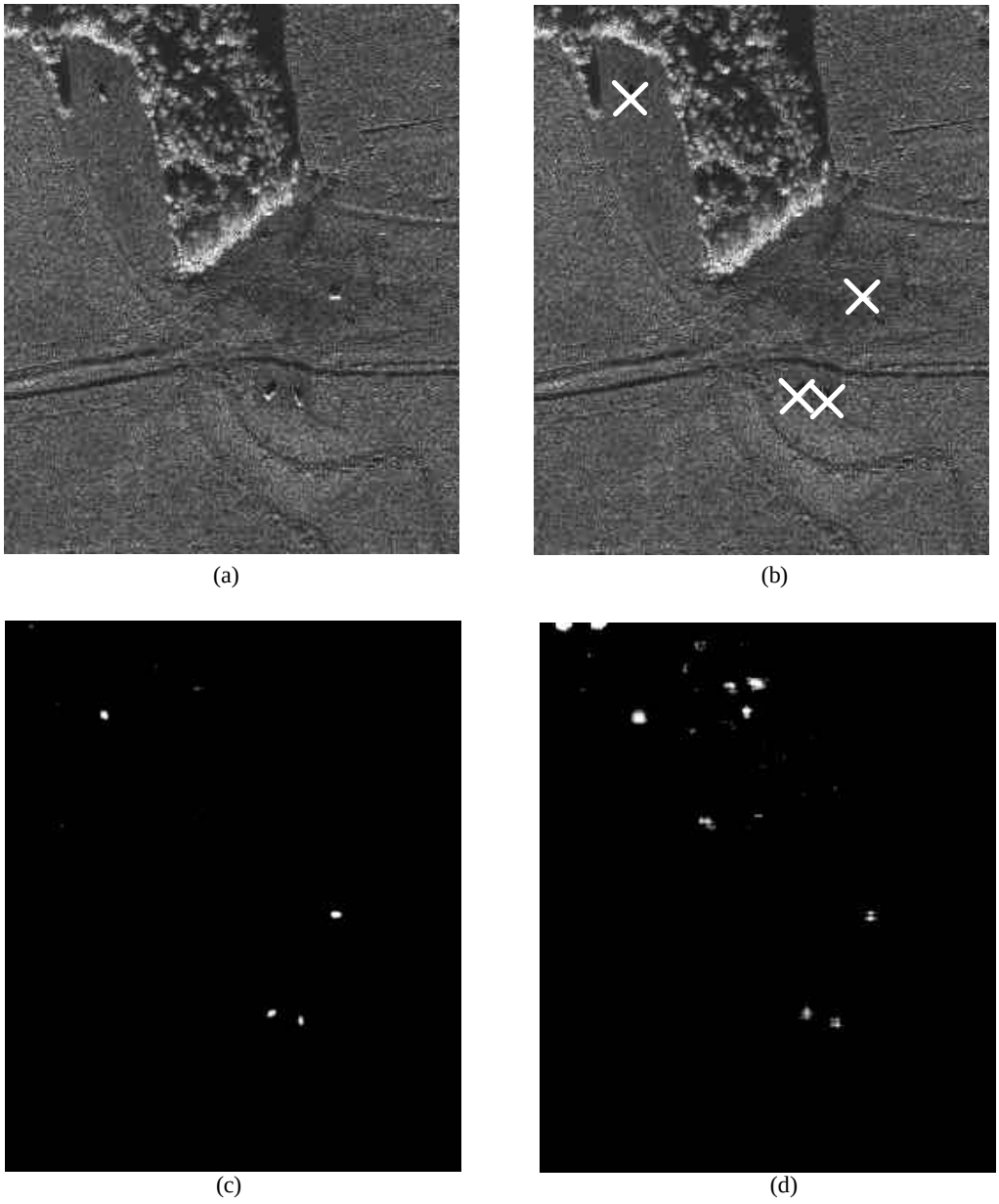


(a) (b)

(c) (d)

(a) Original image, (b) original image in which targets are marked, (c) feature map extracted using the proposed technique, (d) EF feature map

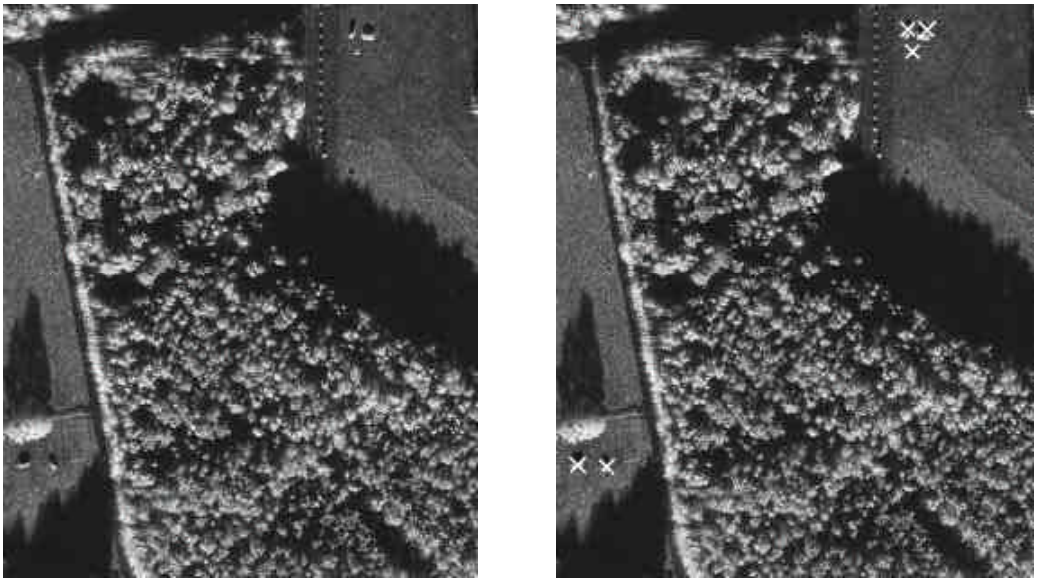
Figure 4.1: Experimental Result 1



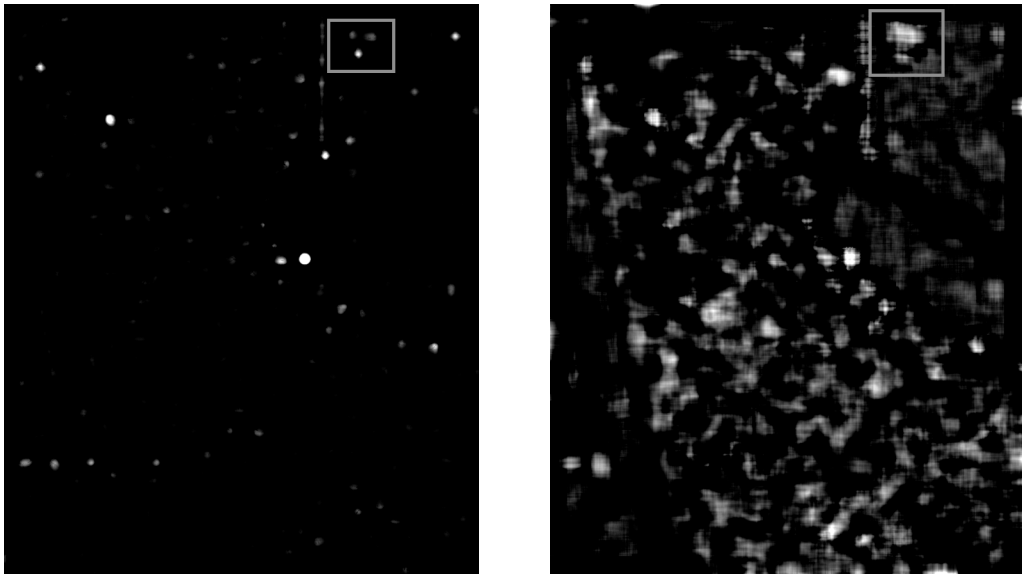
(a) (b) (c) (d)

(a) Original image, (b) original image in which targets are marked, (c) feature map extracted using the proposed technique, (d) EF feature map

Figure 4.2: Experimental Result 2



(a) (b)



(c) (d)

(a) Original image, (b) original image in which targets are marked, (c) feature map extracted using the proposed technique, (d) EF feature map

Figure 4.3: Experimental Result 3

Chapter 5

Conclusions

A new feature extraction technique has been presented for target detection. This work was concentrated in the textural aspects of detection. It is of course understood that additional, non-textural features can be used in order to achieve an improved target detection performance. The author presents a new improved feature extraction scheme based on fractal dimension, and discusses why this is a promising feature. The presented results mostly illustrate the feature's potentials for detecting targets and distinguishing them from clutter. It can be observed that the proposed feature gives a relatively clearer indication of the targets' locations compared to the EF feature.

Future work includes a more thorough comparison between the proposed and other features, including the EF and CFAR features, by employing specific thresholds and classification techniques, and by using a significantly larger number of examples. In this future work, a statistical analysis will be defined and applied to both AR and EF feature maps. In addition, the target detection performance using a mixture of various features will be investigated. The effect of using roughness features extracted in different directions for exploiting the directional target characteristics will be further investigated. As mentioned earlier, roughness features can be extracted in several directions without having a significant computational overhead.

Bibliography

[1] D. Charalampidis and T. Kasparis. Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Transactions on Image Processing*, 11(8):825–837, August 2002.

[2] L. M. Novak, S. D. Halversen, G. J. Owirka, and M. Hiett. Effects of polarization and resolution on SAR ATR. *IEEE Transactions on Aerospace and Electronic Systems*, 33:102–116, January 1997.

[3] G. B. Goldstein. False alarm regulation in log-normal and Weibull clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 9(1):84–92, 1973.

[4] S. Kuttikkad and R. Chellappa. Non-gaussian CFAR techniques for target detection in high resolution SAR images. *IEEE ICIP 94*, pages 910–914, November 1994.

[5] R. Rifkin. Analysis of CFAR performance in Weibull clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 30(2):315–329, April 1994.

[6] V. Anastassopoulos and G.A. Lampropoulos. Optimal CFAR detection in Weibull clutter. *IEEE Transactions on Aerospace and Electronic Systems*, 31(1):52–64, January 1995.

[7] C.J. Morgan, L.R. Moyer, and R.S. Wilson. Optimal radar threshold determination in Weibull clutter and Gaussian noise. *IEEE Transactions on Aerospace and Electronic Systems*, 11(3):41–43, March 1996.

[8] L. M. Novak and S. R. Hesse. On the performance of ordered-statistics CFAR detectors. *Proceedings of the 25th Asilomar Conference on Signals*, pages 835–840, 1991.

[9] L.M. Kaplan. Improved SAR target detection via extended fractal features. *Proceedings of SPIE*, 2755:58–69, April 1996.

[10] J.S. Salowe. Very fast SAR detection. *IEEE Transactions on Aerospace and Electronic Systems*, 37(2):66–75, April 2001.

[11] N. S. Subotic, B. J. Thelen, J. D. Gorman, and M. F. Reiley. Multiresolution detection of coherent radar targets. *IEEE Transactions on Image Processing*, 6:21–35, January 1997.

[12] Q. Pham, T. M. Brosnan, and M. J. T. Smith. Multistage algorithm for detection of targets in image data. *Proceedings of SPIE*, 3070:66–75, April 1997.

[13] C.V. Stewart, B. Moghaddam, K.J. Hintz, and L.M. Novak. Fractional brownian motion models for synthetic aperture radar imagery scene segmentation. *Proceedings of IEEE*, 81(10):1511–1522, October 1993.

[14] L.M. Kaplan. Extended fractal analysis for texture classification and segmentation. *IEEE Transactions on Image Processing*, 8:1572–1585, November 1999.

Appendix

Matlab Code

Three different MATLAB .m files are attached:

- FeatObt.m - Algorithm which calculates the average weighted roughness feature described in Chapter 3.
- wavefeat.m - Algorithm which outputs the directional roughness feature described in Chapter 3
- EF.m - Algorithm which implements the EF feature described in Chapter 2

```
% Implementation of the Wavelet/Fractal feature presented in
% Gregory W. Stein's thesis, titled "Target Detection Using a Wavelet
% Based Fractal Scheme

clear all, close all, clc

% Read SAR image
file='Clutter4.bmp';
fprintf('Image Read...\n');
n_rows = 1474;
n_cols = 1784;

% Parameters - Window size, epsilons, and thetas considered
N=9;
e=[7 11 15];
theta = [0 45 90 135];

% Determine roughness feature at a specified scale
s=1.5; % Define scale
for i=1:length(theta)
    [FV(:,:,i) EN(:,:,i) mag] = wavfeat(file,s,N,e,theta(i),n_rows,n_cols);
    fprintf('Angle %d...\n',theta(i));
end
fprintf('Scale Done...\n\n');

% Determine percentage of energy
for i=1:length(theta)
    PER(:,:,i) = EN(:,:,i)./sum(EN,3);
end

% Compute weighted roughness feature
WR = FV.*PER;

% Compute average weighted roughness feature
AF = sum(WR,3)/length(theta);

% Create omni-directional wavelet - Gabor - Filter optimization
N=25;
x=[-N:N]; x=repmat(x,[2*N+1 1]);
y=[-N:N]'; y=repmat(y,[1 2*N+1]);
s=12;
Wo = cos(2*pi*0.03*sqrt(x.^2+y.^2)).*exp(-(x.^2+y.^2)/(2*s^2)); % Matched Filter

% Optimize with "matched" filter
AFf=filter2(Wo,AF);

% Display - contrast enhanced
figure(1),imshow(mag/255)
figure(2),imshow(-3*AF);
figure(3),imshow(-AFf/100);
figure(4),imshow(AFf.*AF/20);
```

```
function [FV,FVEng,mag]=wavfeat(filename,s,N,e,ang,n_rows,n_cols)
% [FV FVEng] = wavefeat(filename,s,N,e,ang)
%   filename: image filename
%   s: wavelet scale - scalar
%   N: size of wavelet window
%   e: feature epsilon(s) - vector
%   ang: feature direction - scalar
%   n_rows: number of rows in image - scalar
%   n_cols: number of columns in image - scalar

% Read and display target
mag = double(imread(filename));
mag=mag(1:2:end,1:2:end);

% Create n-exponential wavelet
x=[-N:N]; x=repmat(x,[2*N+1 1]);
y=[-N:N]'; y=repmat(y,[1 2*N+1]);

% Define "root" wavelets
W0 = (-x/s).*exp(-(x.^2+y.^2)/(2*s^2)); % Wavelet 0 degree
W90 = (-y/s).*exp(-(x.^2+y.^2)/(2*s^2)); % Wavelet 90 degree

% Filter image with wavelet, take absolute value
if ang == 0
    F = abs(filter2(W0,mag)); % 0 degree
elseif ang == 90
    F = abs(filter2(W90,mag)); % 90 degree
else
    F = abs(filter2(W0,mag)*cos(ang*(pi/180)) + filter2(W90,mag)*sin(ang*(pi/180))); %↙
Any angle
end

M = 19; % Filter window size for moving average
% Directional roughness feature
for i=1:length(e)
    Msize = 2*e(i)+1;
    maxmask = ones(Msize,Msize);
    maxmask(2:end-1,2:end-1) = 0;
    rnk=sum(sum(maxmask));
    FMax(:,:,i) = log(ordfilt2(F,round(rnk*1),maxmask)-ordfilt2(F,round(rnk*0+1),↙
maxmask)+0.0010);
    FMax(:,:,i) = filter2(ones(M)/(M^2),FMax(:,:,i));
end

% Determine slope
[P Q] = size(mag);
le = reshape(log(e),[1 1 length(e)]); % x-axis
st1 = repmat(le-mean(le,3),[P Q 1]);
st2 = FMax - repmat(mean(FMax,3),[1 1 length(e)]);
slope = sum(st1.*st2,3)./sum((st1.^2),3);

% Output
```

```
FV = slope;
FVEng = F;
```

```
% Implementation of the Extended Fractal Feature from
% "Improved SAR Target Detection via Extended Fractal Features
% by Lance M. Kaplan

% Read and display target;
mag=double(imread('Clutter4.bmp'));

% Down-sample Image ------------------------------------------
I = mag(1:2:end,1:2:end);

% Apply EF Feature -------------------------------------------
D = 7; % Delta
W = 4*D+1; % Filter window size
% FE in x-direction
Ifiltx1 = filter2(ones(W,W),(abs(filter2([1,zeros(1,2*D-2),-1]',I))).^2);
% FE in y-direction
Ifilty1 = filter2(ones(W,W),(abs(filter2([1,zeros(1,2*D-2),-1],I))).^2);

D1 = 2*D; % 2xDelta
% FE in x-direction
Ifiltx2 = filter2(ones(W,W),(abs(filter2([1,zeros(1,2*D1-2),-1]',I))).^2);
% FE in y-direction
Ifilty2 = filter2(ones(W,W),(abs(filter2([1,zeros(1,2*D1-2),-1],I))).^2);

Fx = 0.5*log2(Ifiltx1./Ifiltx2); % Ratio in x-direction
Fy = 0.5*log2(Ifilty1./Ifilty2); % Ratio in y-direction
F = (Fx + Fy)/2; % Average

figure(1),imshow(mag/255)
figure(2), imshow(2*F+0.1) % Display
```

Vita

Gregory W. Stein received his Bachelor of Science in Electrical Engineering from The University of New Orleans in May of 2004. He subsequently entered the Masters program as a teaching assistant in the Fall of 2004. He currently works for Industrial Automation Systems, Inc. as a Systems Integrator, and is in the process of starting his own Pro Audio company. His research interests include image and speech processing, focusing on feature extraction and speech synthesis, respectively. He also spends much of his time designing, building, and repairing solid state and tube electronics.